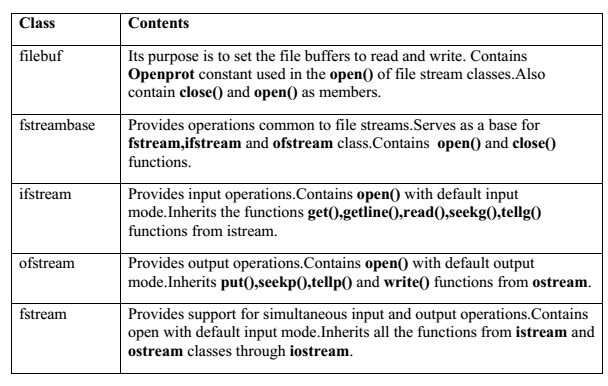
# Working with Files

## Classes for File Stream Operations

The I/O system of C++ contains a set of classes that define the file handling methods. These include **ifstream**, **ofstream** and **fstream**. These classes are derived from **fstreambase** and from the corresponding *iostream*class. These classes designed to manage the disk files, are declared in *fstream* and therefore we must include this file in any program that uses files.



The filename is a string of characters that make up a valid filename for the operating system. It may contain two parts, a primary name and an optional period with extension.

For opening a file, we must first create a file stream and then link it to the filename. A file stream can be defined using the classes **ifstream**, **ofstream**, and **fstream** that are contained in the header file *fstream*. The class to be used depends upon the purpose, that is, whether we want to read data from the file or write data to it. A file can be opened in two ways:

1. Using the constructor function of the class.
2. Using the member function **open( )** of the class.

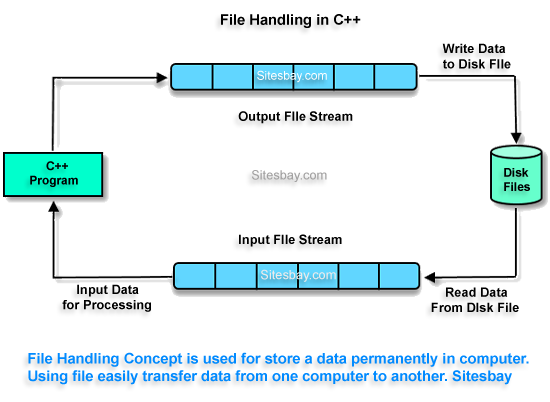
The first method is useful when we use only one file in the stream. The second method is used when we want to manage multiple files using one stream.

# Opening Files Using Constructor

ofstream outfile("FileName") ;   // output only

ifstream infile("FileName") ;    // input only

This creates **outfile** as an **ofstream** object that manages the output stream. This object can be any valid C++ name such as *o\_file*, *myfile* , or *fout*. This statement also opens the file results and attaches is to the output stream **outfile**.



Similarly the above statement declares **infile** as an **ifstream** object and attches it to the file **data** for reading(input).

The program may contain statement like:

outfile<<"total";

outfile<<"sum";

infile>>number;

infile>>string;

we can also use the same file for both reading and writing data.

Program1

……….

……….

ofstream outfile("FileName") ;   // output only

……….

Program2

……….

……….

ifstream infile("FileName") ;   // output only

……….

**NOTE ://** When a file is opened for *writing only*, a new file is created if there is no file of that name. If a file by that name exists already, then its contents are deleted and the file is presented as a clean file.

# Opening Files Using open( )

The function **open( )** can be used to open multiple files that use the same stream object.

Syntax :

**file\_stream\_class** stream\_object ;

stream\_object . **open** (“filename”) ;

Example :

ofstream outfile;

outfile.open(“data1”) ;

….

….

outfile.close( ) ;

outfile.open(“data2”) ;

….

….

outfile.close( ) ;

# Detecting end-of-file

Detection of the end-of-file condition is necessary for preventing any further attempts to read data from the file.

while(stream\_object);

An **ifstream** object, such as **fin** returns a value of 0 if any error occurs in the file operation including the end-of-file condition. Thus , the **while** loop terminates when **fin** returns a value of zero on reaching the end-of-file condition.

There is another approach to detect the end-of-file condition.

if(fin.eof( ) ! = 0 ) {

exit(1) ; }

**eof( )** is a member function of **ios** class. It retuns a non-zero value if the end-of-file (EOF) condition is encountered and a zero otherwise.

# More about Open( ) : File Modes

Open( ) function can take two arguments, the second one for *specifying the file mode*.

stream\_object.open(“filename”,mode) ;

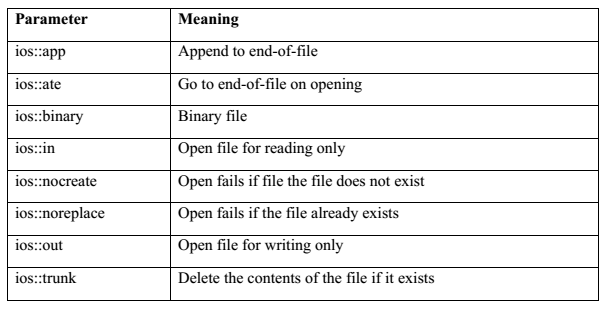
stream\_object.open(“filename”,mode1 | mode2 | mode3) ;

The default values are as follows:

ios::in for ifstream functions meaning open for reading only.

ios::out for ofstream functions meaning open for writing only.

The *file mode* parameter can take on (or more) of such constants defined in the class **ios**.



# Functions for Manipulation of File Pointers

**seekg()** Moves get pointer (input) to a specified location.

**seekp()** Moves put pointer (output) to a specified location.

**Tellg()** Gives the current position of the get location.

**tellp()** Gives the current position of the put pointer.

For example, the statement

Infile.seekg(10) ; moves the file pointer to the byte number 10. The bytes in a file are numbered beginning from zero. Therefore the pointer will be pointing to the 11th byte in the file.

# Specifying the offset

‘Seek’ functions **seekg( )** and **seekp( )** can also be used with two arguments as follows:

seekg(offset, refposition);

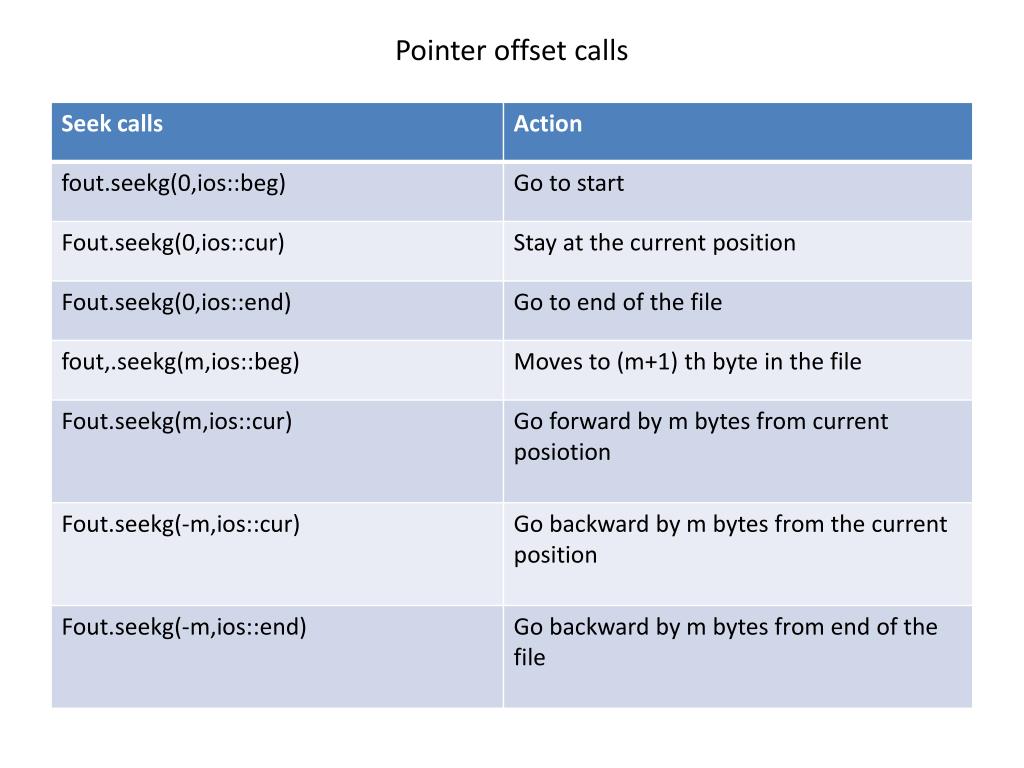
seekp(offset, refposition);

The parameter *offset* represents the number of bytes the file pointer is to be moved from the location specified by the parameter *refposition*. The *refposition* takes one of the following three constants defined in the **ios** class:

ios::beg start of the file

ios::cur current position of the pointer

ios::end End of File



# Sequential Input and Output Operations

The file stream classes support a number of member functions for performing the input and output operations on files. One pair of functions, **put( )** and **get( )**, are designed for handling a single character at a time. Another pair of functions, **write( )** and **read( )**, are designed to write and read blocks of *binary data*.

The function **put( )** writes a single character to the associated stream.

The function **get( )** reads a single character from the associated stream.

# Input and Output in Binary File

The data transfer is usually done using '>>' and <<' operators.

But if you have a class with 4 data members and want to write all 4 data members from its

object directly to a file or vice-versa, we can do that using following syntax :

**To write object's data members in a file :**

// Here file\_obj is an object of ofstream

file\_obj.write((char \*) & class\_obj, sizeof(class\_obj));

**To read file's data members into an object :**

// Here file\_obj is an object of ifstream

file\_obj.read((char \*) & class\_obj, sizeof(class\_obj));

**C++ Programs on File Handling**

Program 60 : Write a C++ program to Create a File.

File Name : C:\TC\BIN\P\_109.CPP

// @author : RONIT

/\*  C++ program to create a File.   \*/

#include<iostream>

#include<stdio.h>

#include<fstream>

using namespace std;

int main()

{

    char ch,filename[10];

    cout<<"Enter File Name with Extension : ";

    cin>>filename;

    ofstream obj(filename);

    while(1){

        ch=cin.get();

        if(ch==EOF)

            break;

        obj.put(ch);

    }

    obj.close();

    return 0;

}

Program 61 : Write a C++ program to Read a File.

File Name : C:\TC\BIN\P\_110.CPP

// @author : RONIT

/\*  C++ program to read a File  \*/

#include<iostream>

#include<stdio.h>

#include<fstream>

using namespace std;

int main()

{

    char ch,filename[10];

    cout<<"Enter File Name with Extension : ";

    cin>>filename;

    ifstream obj(filename);

    while(obj){

        ch=obj.get();

        cout.put(ch);

    }

    obj.close();

    return 0;

}

Program 62 : Write a C++ program to Count the number of vowels, consonants, digits, spaces, punctuation marks in a File.

File Name : C:\TC\BIN\P\_111.CPP

// @author : RONIT

#include<iostream>

#include<stdio.h>

#include<fstream>

#include<string.h>

#include<ctype.h>

using namespace std;

int isvowel(char ch){

    if(ch=='A'||ch=='E'||ch=='I'||ch=='O'||ch=='U')

        return 1;

    else

        return 0;

}

int main()

{

    char ch,filename[10];

    cout<<"Enter File Name with Extension : ";

    cin>>filename;

    int space=0,vowel=0,consonant=0,digit=0,punctuation=0;

    ifstream obj(filename);

    while(obj){

        ch=obj.get();

        cout.put(ch);

        if(isdigit(ch))

            digit++;

        else

        if(ispunct(ch))

            punctuation++;

        else

        if(ch==' ')

            space++;

        else

        if(toupper(ch)>='A'&&toupper(ch)<='Z')

        {

            if(isvowel(toupper(ch)))

                vowel++;

            else

                consonant++;

        }

    }

    obj.close();

    cout<<endl<<endl<<"Number of Spaces : "<<space;

    cout<<endl<<"Number of Vowels : "<<vowel;

    cout<<endl<<"Number of Consonant : "<<consonant;

    cout<<endl<<"Number of Digits : "<<digit;

    cout<<endl<<"Number of Punctuation Marks : "<<punctuation;

    return 0;

}

Program 63 : Write a C++ program to Copy a File and Paste it into another File.

File Name : C:\TC\BIN\P\_112.CPP

// @author : RONIT

#include<iostream>

#include<stdio.h>

#include<fstream>

using namespace std;

int main()

{

    char ch,filename1[10],filename2[10];

    cout<<"Enter Source File Name with Extension : ";

    cin>>filename1;

    cout<<"Enter Destination File Name with Extension : ";

    cin>>filename2;

    ifstream copy(filename1);

    ofstream paste(filename2);

    while(copy){

        ch=copy.get();

        paste.put(ch);

    }

    paste.put(EOF);

    copy.close();

    paste.close();

    cout<<"File Copied Successfully";

    return 0;

}

Program 64 : Write a C++ program to Merge two File into a third File

File Name : C:\TC\BIN\P\_113.CPP

// @author : RONIT

#include<iostream>

#include<stdio.h>

#include<fstream>

using namespace std;

int main()

{

    char ch,filename1[10],filename2[10],filename3[10];

    cout<<"Enter First Source File Name with Extension : ";

    cin>>filename1;

    cout<<"Enter Second Source File Name with Extension : ";

    cin>>filename2;

    cout<<"Enter Destination File Name with Extension : ";

    cin>>filename3;

    ifstream file1(filename1);  // Opening File1

    ofstream merge(filename3);  // Opening Merged File

    while(file1){

        ch=file1.get();         // getting character from File1 into ch

        merge.put(ch);          // putting character into Merged File stored in ch

    }

    file1.close();              // closing File1

    ifstream file2(filename2);  // Opening File2

    while(file2){

        ch=file2.get();         // getting character from File2 into ch

        merge.put(ch);          // putting character into Merged File stored in ch

    }

    file2.close();              // closing File2

    merge.put(EOF);             // appending EOF in Merged File

    merge.close();              // closing Merged File

    cout<<"File Merged Successfully";

    return 0;

}

Program 65 : Write a C++ program to Store Student Data in a Binary File

File Name : C:\TC\BIN\P\_114.CPP

// @author : RONIT

/\*  Write a C++ program to store student data in a File \*/

#include<iostream>

#include<fstream>

#include<ctype.h>

#include<conio.h>

using namespace std;

class Student{

    int roll;

    char name[30];

    int total;

    public:

        void getdata(){

            cout<<"\n Enter Roll : ";

            cin>>roll;

            cout<<"\n Enter Name : ";

            cin>>name;

            cout<<"\n Enter Marks: ";

            cin>>total;

        }

        void putdata(){

            cout<<"\n Roll = "<<roll;

            cout<<"\n Name = "<<name;

            cout<<"\n Marks= "<<total;

        }

};

int main()

{

    fstream myfl;

    myfl.open("student.dat",ios::out|ios::binary);

    char ch;

    Student obj;

    while(1){

        obj.getdata();

        myfl.write((char\*) &obj,sizeof(obj));

        cout<<"\n Continue?(Y/N) ";

        cin>>ch;

        if(toupper(ch)=='N')

            break;

    }

    myfl.close();

    myfl.open("student.dat",ios::in|ios::binary);

    myfl.seekg(0,ios::beg);

    while(myfl.read((char\*) &obj,sizeof(obj)))

        obj.putdata();

    myfl.close();

    return 0;

}

Program 66 : Write a C++ program to Replace all occurrence of MAY with CAN

File Name : C:\TC\BIN\P\_115.CPP

// @author : RONIT

/\*  Description : Replace all occurrence of may with can    \*/

#include<iostream>

using namespace std;

#include<fstream>

#include<conio.h>

#include<string.h>

int main()

{

    char ch,filename[20];

    fstream fobj;

    cout<<"\n Enter File Name to be opened : ";

    cin>>filename;

    fobj.open(filename,ios::in|ios::out);

    while(!fobj.eof())

    {

        ch=fobj.get();

        cout<<ch;       // he may write.

    }

    fobj.clear();

    fobj.seekg(0);  // move read pointer  from beg

    char may[5]="may ",can[5]="can ",word[10];

    int i=0,pos;

    while(fobj)

    {

        ch=fobj.get();

        word[i++]=ch;

        if(ch==' ')

        {

            word[i++]='\0';

            if(strcmp(word,may)==0)

            {

                pos=fobj.tellg();

                fobj.seekp(pos-4,ios::beg);

                for(int j=0;j<4;j++)

                {

                    fobj.put(can[j]);

                }

            }

            word[0]='\0';

            i=0;

        }

    }

    fobj.clear();

    fobj.seekg(0);

    cout<<endl<<"After Replacing "<<endl;

    while(fobj)

    {

        ch=fobj.get();

        cout<<ch;

    }

    fobj.clear();

    fobj.close();

    return 0;

}

Program 67 : Write a menudriven C++ program to create a Student Database.

Include the operations like : append, edit, search, listing, delete.

File Name : C:\TC\BIN\P\_116.CPP

// @author : RONIT

#include<iostream>

using namespace std;

#include<fstream>

#include<string.h>

#include<ctype.h>

#include<stdlib.h>

#include<iomanip>

#include<stdio.h>

#include<conio.h>

class Student{

    int roll;

    char name[30];

    int mark;

    public:

        void getdata(){

            cout<<"\n Enter Roll : ";

            cin>>roll;

            cout<<" Enter Name : ";

            cin>>name;

            cout<<" Enter Marks: ";

            cin>>mark;

        }

        void putdata(){

            cout<<endl<<setw(5)<<roll<<setw(10)<<name<<setw(5)<<mark<<setw(10);

        }

        void WriteFile(char\*);

        void ShowFile(char\*);

        void EditFile(int,char\*);

        void SearchFile(int,char\*);

        void DeleteFileData(int,char\*);

};

void Student::WriteFile(char \*filename){

    fstream obj;

    obj.open(filename,ios::app|ios::binary);

    obj.write((char \*)this,sizeof(\*this));

    obj.close();

}

void Student::ShowFile(char \*filename){

    fstream obj;

    obj.open(filename,ios::in|ios::binary);

    if(!obj)

        cout<<"\n File not found";

    else{

        obj.read((char \*)this,sizeof(\*this));

        if(obj.eof())

            cout<<"\n No Records Found.";

        while(!obj.eof()){

            putdata();

            obj.read((char \*)this,sizeof(\*this));

        }

        obj.close();

    }

}

void Student::SearchFile(int rollno,char \*filename){

    fstream obj;

    int flag=0;

    obj.open(filename,ios::in|ios::binary);

    if(!obj)

        cout<<"\n File not found";

    else{

        obj.read((char \*)this,sizeof(\*this));

        while(!obj.eof()){

            if(rollno==roll){

                putdata();

                flag=1;

            }

            obj.read((char \*)this,sizeof(\*this));

        }

        obj.close();

        if(flag==0)

            cout<<"\n No Records found";

    }

}

void Student::EditFile(int rollno,char \*filename){

    fstream obj;

    obj.open(filename,ios::in|ios::out|ios::ate|ios::binary);

    obj.seekg(0);

    obj.read((char \*)this,sizeof(\*this));

    while(!obj.eof()){

        if(rollno==roll){

            getdata();

            obj.seekp(obj.tellp()-sizeof(\*this));

            obj.write((char \*)this,sizeof(\*this));

        }

        obj.read((char \*)this,sizeof(\*this));

    }

}

void Student::DeleteFileData(int rollno,char \*filename){

    fstream obj,temp;

    obj.open(filename,ios::in|ios::binary);

    if(!obj)

        cout<<"\n File not found";

    else{

        temp.open("TemporaryFile.dat",ios::out|ios::binary);

        obj.read((char \*)this,sizeof(\*this));

        while(!obj.eof()){

            if(rollno!=roll)

                temp.write((char \*)this,sizeof(\*this));

            obj.read((char \*)this,sizeof(\*this));

        }

        temp.close();

        obj.close();

        remove(filename);

        rename("TemporaryFile.dat",filename);

    }

}

int main()

{

    Student s1;

    int choice,temp;

    char filename[20];

    cout<<"Enter File Name to be opened : ";

    cin>>filename;

    do{

        cout<<"\n\n Menu ";

        cout<<"\n 1. Add Data";

        cout<<"\n 2. Edit Data";

        cout<<"\n 3. Search Data";

        cout<<"\n 4. Display Data";

        cout<<"\n 5. Delete Data";

        cout<<"\n 0. Exit \t\t";

        cin>>choice;

        switch(choice){

            case 1 : // Append

                        s1.getdata();

                        s1.WriteFile(filename);

                        break;

            case 2 : // Edit

                        cout<<"\n Enter Roll No to be edited : ";

                        cin>>temp;

                        s1.EditFile(temp,filename);

                        break;

            case 3 : // Search

                        cout<<"\n Enter Roll No to be searched : ";

                        cin>>temp;

                        s1.SearchFile(temp,filename);

                        break;

            case 4 : // Display

                        s1.ShowFile(filename);

                        break;

            case 5 : // Delete

                        cout<<"\n Enter Roll No to be deleted : ";

                        cin>>temp;

                        s1.DeleteFileData(temp,filename);

                        break;

            case 0 : exit(0);

            default : cout<<"\n Invalid Choice ";

                        break;

        }

    }while(choice!=0);

    return 0;

}

…………………End of File………..